**1 import java.util.\*;**

**2 import java.io.\*;**

**3 // Student Starter**

**4 public class ContactList**

**5 {**

**6 /\*\***

**7 Contact list file name**

**8 \*/**

**9 private String filename;**

**10**

**11 /\*\***

**12 ContactList constructor accepts a String parameter**

**13 \*/**

**14 public ContactList(String inFileName)**

**15 {**

**16 filename = inFileName;**

**17 }**

**18**

**19 /\*\***

**20 3) add a new record to the file. Open the file for**

**21 writing in append mode(there is a FileWriter constructor**

**22 with the appropriate parameters).**

**23 a) prompt the user to enter data for each field in the record.**

**24 Each field is a String. The last name is required. If the**

**25 last name is the empty string(""), return to the menu.**

**26 b) when the user has completed entering data(i.e., all the**

**27 fields have been prompted), re-display the user choices**

**28 c) do not overwrite existing data**

**29 \*/**

**30 public void new\_record()**

**31 {**

**32 /\***

**33 Prompt for data:**

**34 Last name**

**35 First name**

**36 Phone**

**37 \*/**

**38 //Craete a scanner object**

**39**

**40 //prompt for the last name**

**41**

**42 //input the last name**

**43**

**44 //the Last\_name must not be empty**

**45 if( )**

**46 {**

**47 //get the first name and the phone**

**48**

**49 //create the output string**

**50**

**51 //delare variables to hold file types**

**52**

**53 //try to open the file for writing - append the data**

**54 try**

**55 {**

**56**

**57 }**

**58 catch(IOException ioe)**

**59 {**

**60 System.out.println("new\_record: Exception opening the file for writing");**

**61 }**

**62 //try to wrtie the data**

**63 try**

**64 {**

**65**

**66 }**

**67 catch(IOException ioe)**

**68 {**

**69 System.out.println("new\_record: Exception writing to the file");**

**70 }**

**71 //try to close the file**

**72 try**

**73 {**

**74**

**75 }**

**76 catch(IOException ioe)**

**77 {**

**78 System.out.println("new\_record: Exception closing the file");**

**79 }**

**80 }//end of test of Last\_name**

**81 }//end of new\_record**

**82**

**83 /\*\***

**84 2)display all last names and first names in the file.**

**85 Open the file for reading, read each record and**

**86 display the field values.**

**87 a) display all the lastName, firstName paired fields in**

**88 the file; display with the format lastName, firstName**

**89 b) when all records have been displayed, display the**

**90 record count - the record count is the number of records**

**91 read and should equal the number of records in the file**

**92 c) after all the records and the count have been displayed,**

**93 display the user choices**

**94 \*/**

**95 public void display\_names()**

**96 {**

**97 //delare variables to hold file types**

**98**

**99 //try to open the file for reading**

**100 try**

**101 {**

**102**

**103 }**

**104 catch(IOException ioe)**

**105 {**

**106 System.out.println("display\_names: Exception opening the file");**

**107 }**

**108**

**109 /\***

**110 Try to read each record and display the field values.**

**111 a) display all the lastName, firstName paired fields in**

**112 the file; display with the format lastName, firstName**

**113 count each record that is read**

**114 \*/**

**115 int counter = 0; //record counter**

**116 try**

**117 {**

**118 //read the first record**

**119 String line = br.readLine();**

**120 //while the record is not null, display the record, count the record**

**121**

**122 }**

**123 catch(IOException ioe)**

**124 {**

**125 System.out.println("display\_names: Exception reading the file");**

**126 }**

**127**

**128 //try to close the file**

**129 try**

**130 {**

**131**

**132 }**

**133 catch(IOException ioe)**

**134 {**

**135 System.out.println("display\_names: Exception closing the file");**

**136 }**

**137 //dislay a count of the records read**

**138**

**139 }//end of display\_names**

**140**

**141 /\*\***

**142 1) search an address file for a particular last name**

**143 and then display the Last name, the first name, and**

**144 the phone for each match**

**145 2) display the count of records which match the last name**

**146 \*/**

**147 public void search(String LastName)**

**148 {**

**149 //delare variables to hold file types**

**150**

**151 //try to open the file for reading**

**152 try**

**153 {**

**154**

**155 }**

**156 catch(IOException ioe)**

**157 {**

**158 System.out.println("search: Exception opening the file");**

**159 }**

**160**

**161 /\*\***

**162 1)search an address file for a particular last name**

**163 and then display the Last name, the first name, and**

**164 the phone for each match**

**165 2)display the count of records which match the last name**

**166**

**167 \*/**

**168 int counter = 0;**

**169**

**170 try**

**171 {**

**172 //read the first record**

**173**

**174 /\***

**175 try to read each record; if the value of the Last\_name**

**176 field equals the value, create a counter to count the**

**177 number of matching records**

**178 \*/**

**179**

**180 }**

**181 catch(IOException ioe)**

**182 {**

**183 System.out.println("search: Exception reading the file");**

**184 }**

**185**

**186 // try to close the file**

**187 try**

**188 {**

**189**

**190 }**

**191 catch(IOException ioe)**

**192 {**

**193 System.out.println("search: Exception closing the file");**

**194 }**

**195 //dislay a count of the records found**

**196**

**197 }//end of search**

**198 }//end of class**